REMARKS

Claims 1-4 and 6-19 were pending.

Claims 1-4 and 6-19 are rejected.

Claims 12 and 20 are cancelled.

Claims 1, 6, 11, and 13 have been amended.

Claims 1-4 and 6-11, and 13-19 are pending.

Claim Amendments

Claims 1, 6, 11, and 13 have been amended. Support for the amendments can be found in the application as filed, for example, on pages 14-16, 21-22, and FIG. 2. No new matter has been added.

Claim Rejections – 35 USC §102

Claims 11-12 are rejected under 35 USC 102(b) as being anticipated by Andrew S. Tanenbaum, "Computer Networks," Third Edition 1996 ("Tanenbaum").

Claim 11

Claim 11 includes "analyzing an incoming data packet according to a plurality of sets of parameters, wherein the sets of parameters analyzed depends upon a type of service access point from which the data packet came, each set of parameters includes a priority, and the sets of parameters are used in analyzing the data packet according to an order of the priorities of the sets of parameters." Claim 11 as amended includes elements similar to cancelled claim 12. Accordingly, the rejection of claim 12 will be addressed in this discussion

The Examiner has identified a set of parameters as the parameters of an IP packet header and the data packet as an IP packet. The Examiner has indicated that these IP parameters are analyzed at a network layer. Using the IP packet header parameters, which the Examiner cited as a set of parameters, as an example, there is no indication that there is a priority associated with those IP packet header parameters. Furthermore, there is no indication of another set of IP packet header parameters also has a priority, and is also used to analyze a data packet according to that priority.

In the Advisory Action dated January 24, 2008, the Examiner states that the "claim language of claim 12 does not specify that the priority parameter is associated with other fields

(parameters)." The Examiner is reminded that it was the Examiner's interpretation of a set of parameters as IP packet header parameters. The Applicant was using the Examiner's interpretation to illustrate the flaws of the Examiner's argument.

Moreover, when rejecting claim 12 the Examiner cited the "precedence field" which allows routers to make choices between various links. Tanenbaum, p. 414. However, it is the order of the analysis of the sets of parameters that is determined by the priority, not the order of the packets when routed.

Accordingly, Tanenbaum does not teach each and every element of claim 11.

Claim Rejections – 35 USC §103

Claims 1-4, 14-19 are rejected under 35 USC 103(a) as being unpatentable over W. Richard Stevens, "UNIX Network Programming," 1990, ("Stevens"), in view of US Pub. No. 2003/0103521 to Raphaeli ("Raphaeli")

Claim 1

Claim 1 includes "receiving application data in a protocol layer from an application in a device through a service access point, the service access point being one of a plurality of service access points of the protocol layer; classifying the application data in the protocol layer as internet protocol (IP) based or non-IP based according to the associated service access point after receiving the application data through the service access point; determining in the protocol layer if a connection exists for the application data in response to the classification of the application data."

Thus, the classification and the determination if a connection exists operates in response to application data received through the service access point in a protocol layer. That is, the receiving, classification, and determination are all on the system side or downstream side of the service access point.

In contrast, the Examiner cited various Berkeley Sockets system calls such as socket, connect, listen, accept, and the like. As these are system calls, they are on the application side of a socket, which was interpreted by the Examiner as the service access point. Thus, Stevens does not describe the operation of the system, rather the interface to the system through he system calls.

Moreover, assuming for the sake of argument that the Berkeley Sockets system calls are used at the system level such as in accessing a lower protocol layer, Stevens still only describes the interface to a protocol layer, not the implementation. There is still no suggestion of using the family variable of a first socket call, cited by the Examiner as the classification of IP or non-IP, in another lower level socket related call. For example, if the family is set in an application level socket call, there is no suggestion that it is propagated to a lower level connect call as suggested by the Examiner.

Furthermore, Raphaeli is silent on IP, instead describing a powerline MAC layer. Thus, it does not suggest using a classification as IP or non-IP in determining if a powerline MAC connection exists.

In addition, claim 1 recites "classifying the application data in the protocol layer as internet protocol (IP) based or non-IP based according to the associated service access point after receiving the application data through the service access point." Thus, the classification as IP or non-IP occurs after the application data has been received through the service access point.

As a result, the combination of Stevens and Raphaeli does not teach or suggest each and every element of claim 1.

Claims 14-16

Each of claims 14-16 includes accessing a classification table for a mapping of the service access point to a connection identifier. Dependent claims 15 and 16 add additional elements for the mapping. Since the classification table includes a mapping of the service access point and the connection identifier, the service access point and the connection identifier are distinct. In parent claim 1, application data is received through a service access point prior to determining if a connection exists for the application data. Thus, the connection is distinct from the service access point.

In contrast, the Examiner is interpreting fields of a cited socket descriptor as both the service access point and the connection identifier. Stevens states that all elements of the 5-tuple must be specified before the socket descriptor is of any real use. Stevens, p. 269. Thus, the socket descriptor must be fully described prior to receiving application data through the socket.

The Examiner has give no rational reason why a connection created between the service access point and a remote service access point would be checked to see if it exists after the application data was received through the service access point.

Moreover, in claim 15 at least one of an IP address, a port number, and a type of service field is used with the service access point in the mapping. In claim 16, both an IP address and a port number are used with the service access point in the mapping. The Examiner's interpretation of the socket descriptor as both a service access point, at least one of an IP address, a port number, and a type of service field, and a connection identification makes the clause "at least one of an IP address, a port number, and a type of service field" meaningless.

Note that removing the addresses of the socket descriptor leaves only the protocol field and the process fields. This is in direct conflict to Tanenbaum where the transport service access point (TSAP) is described with reference to the Internet as an IP address and a local port. Tanenbaum, p. 489. In other words, the IP addresses are part of the service access point. One skilled in the art would understand that the socket descriptor as a whole describes the socket. Thus an additional mapping of at least one of an IP address, a port number, and a type of service field and a connection identifier are needed. This additional mapping is not suggested by the combination of Stevens and Raphaeli.

As a result, the combination of Stevens and Raphaeli does not teach or suggest each and every element of claims 14-16.

<u>*Claims 17-19*</u>

Claims 17-19 include comparison of the application data with particular classifier rules for a match. The Examiner cites the 5-tuple socket descriptor as values to be compared. However, nowhere in the cited section of Stevens is the socket descriptor described as being compared. Moreover, there is no suggestion of comparing the socket descriptor to the application data. , as described above the service access point adds what is in the header.

For example, assume for the sake of argument that application data is received through an IP socket. If some IP address is not part of the application data, then it would not be compared against an IP address in the socket descriptor. Alternatively, as described above, the protocol layer adds a header to the data. Thus, an IP header would be added to data passing through an IP service access point. The IP address would be identical to the IP address in the socket descriptor. There would always be a match. One skilled in the art is not going to add unnecessary comparisons where there would always be a match as described by the Examiner.

Moreover, in claim 19, the application data is only compared to "at least one destination address within the at least one classifier rule." The at least one classifier rule was introduced in

claim 17 for providing a connection where there is a match between the application data and the at least one rule. Hence, when providing a connection for the application data, the application data is compared with only at least one destination address before a connection is provided.

In contrast, the Examiner cited the comparison at a receiving end socket. If the data has arrived at the receiving end, then a connection was already provided for the data to get to the receiving end. In addition, the Examiner has given no rational reason why the same classifier rule used at the originating end to identify a connection is used again at the receiving end.

As a result, the combination of Stevens and Raphaeli does not teach or suggest each and every element of claims 17-19.

Claim 6

Claims 6-10 are rejected under 35 USC 103(a) as being unpatentable over Tanenbaum in view of Stevens.

Claim 6 includes "receiving an incoming data packet from an application on a device at one of a plurality of service access points of a protocol layer; and classifying the data packet in the protocol layer according to the service access point and at least one rule, causing the packet to be associated with a connection established at an interface between the first protocol layer and a second protocol layer, wherein the second protocol layer is a lower level protocol layer." That is, the classification occurs on the system side of a service access point.

The Examiner cited the SOCKET and CONNECT primitives to show receiving the incoming data packet and associating the data packet with a connection. As can be seen in the cited Fig. 6-6, the SOCKET primitive creates a communication end point and the CONNECT primitive attempts to establish a connection. Tanenbaum, p. 487. Creating a communication end point (SOCKET) is not receiving an incoming data packet. Similarly, attempting to establish a connection (CONNECT) is not associating a packet with a connection.

Moreover, connections as used in Tanenbaum are always between end point to end point, not from an end point up the protocol layers to an application. See Tanenbaum, p. 490. Assuming for the sake of argument that the Examiner meant to say that the RECEIVE primitive was receiving an incoming data packet, the characterization of the application data is in direct conflict with other claim elements.

For example, if the data packet is received through a local socket, then there is no suggestion of routing the packet to the connection and transmitting the data. Tanenbaum and

Stevens both describe the interface to a network layer looking into the network, not processing that occurs on received packets.

In the Advisory Action dated January 24, 2008, the Examiner stated "Instead of argues specifically on the reason why Examiner's interpretation, Applicant simply states "there is no suggestion of routing the packet"." The Examiner is reminded that the only primitives that the Examiner cited for receiving data was the SOCKET primitive and possibly the CONNECT primitives. As described above, the SOCKET and CONNECT primitives <u>do not receive data</u>.

To move prosecution along, the Applicant assumed that the Examiner actually meant the RECEIVE parameter, where data can actually be transferred. The above discussion was to point out that the RECEIVE primitive moves data in the opposite direction as described in claim 6. That is, the RECEIVE primitive moved data up towards the application, rather than from the application towards the system level.

As a result, the combination of Stevens and Tanenbaum does not teach or suggest each and every element of claims 6-10.

<u>Claim 13</u>

Claim 13 is rejected under 35 USC 103(a) as being unpatentable over Tanenbaum. Claim 1 includes "analyzing a set of parameters in an incoming data packet, wherein the set of parameters analyzed depends upon a type of service access point from which the data packet came."

The Examiner cited option negotiation to show analyzing a set of parameters of an incoming packet. There is no suggestion that an incoming data packet is analyzed for the quality of service parameters described in Fig. 6-2. Tanenbaum, p. 482-483. The Examiner claimed that the option negotiation applies to socket parameters. However, the Examiner has given no rational reason why an incoming data packet is analyzed for negotiation of quality of service parameters of a connection, or even why socket parameters would be in an incoming data packet.

As a result, the Tanenbaum does not teach or suggest each and every element of claim 13.

For the foregoing reasons, reconsideration and allowance of claims 1-4 and 6-11, and 13-20 of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,

MARGER JOHNSON & McCOLLOM, P.C.

Derek Meeker
Reg. No. 53,313

MARGER JOHNSON & McCOLLOM, P.C.
210 SW Morrison Street, Suite 400
Portland, OR 97204
503-222-3613
Customer No. 46404